Designer Private Edition Guide

Configure Designer

2/18/2026

Contents

Learn how to configure Designer.

Related documentation:

-
-
-

RSS:

- For private edition

Deployment configuration settings (Helm values)

The following sections provide information on the various settings that have to be configured in Designer and DAS. The configuration settings listed below will be used during the deployment of Designer and DAS. That is, these settings will be used during initial deployment/upgrade. These settings can be configured in the **values.yaml** Helm file.

For more information about how to override Helm chart values, see Overriding Helm chart values in the *Setting up Genesys Multicloud CX Private Edition* guide.

> ### Important
> Depending on the Kubernetes platform or the container orchestration platform that you are deploying Designer on, you might have to carry out some additional steps specific to that platform. For more information, navigate to the required topic in the **Kubernetes platform specific information** section on the About page.

Designer deployment settings

The following table provides information on the Designer deployment settings. These settings are configured in the **designer-values.yaml** file.

Parameter	Description	Mandatory?	Default Value
`designer.deployment.replicaCount`	Number of service instances to be created.	Mandatory	2
`designer.deployment.maxReplicaCount`	The maximum number of replicas to be created. It is	Optional	`10`

	recommended to configure this setting if auto-scaling is used.		
`designer.deployment.strategy`	The deployment strategy to follow. This determines which type of resources are deployed. Valid values are: `rollingupdate`, `blue-green`, `blue-green-volume`, `blue-green-ingress`, `grafana`. • **rollingupdate** - default Kubernetes update strategy where resources will be updated using the rolling upgrade strategy. • **blue-green** - for deploying and upgrading the Designer service using the blue-green strategy. • **blue-green-volume** - for the blue/green upgrade, this is to create a Persistent Volume Claim (PVC) for the very first time. • **blue-green-ingress** - for the blue/green upgrade, this is to create an ingress for the first time and update the ingress during a service cutover. • **grafana** - for deploying the Grafana dashboard.	Mandatory	`rollingupdate`
`designer.deployment.color`	This is to deploy/ upgrade the Designer service in a blue-green upgrade strategy. Valid values are: `blue`, `green`.	Optional	
`designer.deployment.type`	This is to specify the	Optional	`Deployment`

	type of deployment. Valid value: `Deployment`.		
`designer.image.registry`	The registry that the organization uses for storing images.	Mandatory	
`designer.image.repository`	Docker repository that contains the images for Designer.	Mandatory	
`designer.image.tag`	Designer image version.	Mandatory	9.0.110.07.7
`designer.image.PullPolicy`	Designer image pull policy (imagePullPolicy). Valid values: `Always, IfNotPresent, Never`. • **Always** - always pull the image. • **IfNotPresent** - pull the image only if it does not already exist on the node. • **Never** - never pull the image.	Mandatory	`IfNotPresent`
`designer.image.imagePullSecrets`	Secret name containing credentials for authenticating access to the Docker repository.	Mandatory	
`designer.volumes.workspacePv.create`	`true` if a persistent volume for the Designer workspace must be created. This is used in case of static volume provisioning, where, the PV is created and then the PVC is bound to the specified PV. Currently, support to create PV only for Azure files (SMB) and NFS is present in the helm chart.		`false`
`designer.volumes.workspacePv.type`	Supports two types: `nfs` - Creates an NFS PV provided you have an NFS server/file share set up already. `azurefiles-smb` - Creates a PV for pre-existing SMB type Azure fileshares.		
`designer.volumes.workspacePv.name`	Name of the PV to be created. For example,		

	`designer-workspace-pv`.		
`designer.volumes.workspacePv.storage`	Size of the PV to be created. For example, `5Gi`.		
`designer.volumes.workspacePv.storageClass`	The storage class associated with the PV. For static volume provisioning to occur as expected, it is highly recommended to provide `""` (intentional empty double quotes) or any distinct storage class name that does not exist already.		
`designer.volumes.workspacePv.mountOptions`	Mount options to be given to the PV. Note: Mount options differ according to the underlying storage type used, that is, NFS or SMB. Using the same set of `mountOptions` with different storage types leads to volume mount errors.		
`designer.volumes.workspacePv.server`	The IP address or FQDN of the NFS server. Note: This field is only applicable for `nfs` type PVs.		
`designer.volumes.workspacePv.path`	The exported path from the NFS server. Note: This field is only applicable for `nfs` type PVs.		
`designer.volumes.workspacePv.shareName`	The azure fileshare name for which the PV must be created. Note: This field is only applicable for `azurefiles-smb` type PVs.		
`designer.volumes.workspacePv.createSecret`	`true` if secret with data to authenticate the Azure storage account must be created. Can be `false` if the secret is manually created. Note: This field is only applicable for `azurefiles-smb` type PVs.		
`designer.volumes.workspacePv.secretName`	The name to be given to		

	the secret created with the `designer.volumes.workspacePv.createSecret` field. For example, `designer-storage-secret`). Note: This field is only applicable for `azurefiles-smb` type PVs.		
`designer.volumes.worksp`	Base64 encoded name of the storage account. This goes in the secret created with `designer.volumes.workspacePv.createSecret`. Note: This field is only applicable for `azurefiles-smb` type PVs.		
`designer.volumes.worksp`	Base64 encoded access key of the storage account. This goes in the secret created with `designer.volumes.workspacePv.createSecret` Note: This field is only applicable for `azurefiles-smb` type PVs.		
`designer.volumes.workspacePvc.create`	If a persistent volume is to be created, this value has to be `true`.	Mandatory	`true`
`designer.volumes.workspacePvc.type`	The type of the volume provisioning to use: `static` - This type is used when a PV has been created either by using the helm values in `designer.volumes.workspacePv` or bmanually and the workspace PVC must be bound to it. `dynamic` - This type is used when a configured storage class will dynamically allocate a PV to the workspace PVC.	Mandatory	`dynamic`
`designer.volumes.workspacePvc.mountPath`	The path where the workspace volume is to be mounted inside the Designer container.	Mandatory	`/designer/workspace` Note: This is not a customizable value. The value MUST be `/designer/workspace` for the proper functioning of Designer.
`designer.volumes.workspacePvc.claim`	Persistent volume claim name for the workspace.	Mandatory	`designer-managed-disk`

`designer.volumes.workspacePvc.claimSize`	Size of the persistent volume claim for the workspace. The persistent volume must be equal to or greater than this size.	Mandatory	
`designer.volumes.workspacePvc.storageClass`	storageClassName provided in the persistent volume that is created for the Designer workspace (example, **nfs**).	Mandatory	
`designer.volumes.workspacePvc.volumeName`	The PV's name to which the PVC must be bound (applicable only when `designer.volumes.workspacePvc.type` is `static`).		
`designer.volumes.logsPvc.create`	If a PVC volume is to be created, this value has to be `true`, else `false`.	Mandatory	`true`
`designer.volumes.logsPvc.type`	The type of volume provisioning to use: `static` - This type is used when a PV has been created and PVC logs must be bound to it. `dynamic` - This type is used when a configured storage class will dynamically allocate a PV to the PVC logs. Note: The helm charts only have support for creating static PVs for the PVC workspace. For PVC logs, it is recommended to make use of dynamic provisioning and let the storage class do the PV allocation.		
`designer.volumes.logsPvc.mountPath`	The path where the Designer logs volume is to be mounted inside the Designer container.	Mandatory	`/designer/logs` Note: This is not a customizable value. The value MUST be `/designer/logs` for the proper functioning of Designer.
`designer.volumes.logsPvc.claim`	Persistent volume claim name for logs.	Mandatory	`designer-logs`
`designer.volumes.logsPvc.claimSize`	Size of the persistent volume claim for the Designer logs. The persistent volume must be equal to or greater than this size.	Mandatory	
`designer.volumes.logsPvc.storageClass`	storageClassName	Mandatory	

	provided in the persistent volume that is created for the Designer logs (example, **nfs**). Note: In case of static volume provisioning, this field must match with the storage class of the PV. If the PV does not have a storage class, then it is mandatory to provide "" for this field in the helm values. Otherwise, static volume provisioning will not occur as expected.		
`designer.volumes.logsPvc.pvName`	The PV's name to which the PVC must be bound (applicable only when `designer.volumes.logsPvc.type` is static).		

`designer.podVolumes`	Log and workspace persistent volume claim names and name of the volumes attached to the pod.	Mandatory	`designer:`
			`  podVolumes:`
			`    - name: designer-pv-volume`
			`      persistentVolumeClaim:`
			`        claimName: designer-managed-disk`
			`    - name: designer-log-volume`
			`      persistentVolumeClaim:`
			`        claimName: designer-logs`

`designer.volumeMounts`	Name and mount path of the volumes to be attached to the Designer pods.	Mandatory	`volumeMounts:`
			`    - name: designer-pv-volume`
			`      mountPath: /designer/workspace`
			`    - name: designer-log-volume`
			`      mountPath: /designer/logs`

`designer.livenessProbe.path`	Designer liveness probe API path.	Mandatory	`/health`
`designer.livenessProbe.containerPort`	Port running the container.	Mandatory	`8888`
`designer.livenessProbe.startupDelay`	The liveness probe will be started after a given delay as specified here.	Mandatory	`20`
`designer.livenessProbe.checkInterval`	The interval between each liveness probe request.	Mandatory	`5`

`designer.livenessProbe.failureCount`	Number of liveness probe failures after which to mark the container as unstable or restart.	Mandatory	`5`
`designer.readinessProbe.path`	Designer readiness probe API path.	Mandatory	`/health`
`designer.readinessProbe.containerPort`	Port running the container.	Mandatory	`8888`
`designer.readinessProbe.startupDelay`	The readiness probe will be started after a given delay as specified here.	Mandatory	`20`
`designer.readinessProbe.checkInterval`	The interval between each readiness probe request.	Mandatory	`5`
`designer.readinessProbe.failureCount`	Number of readiness probe failures after which to mark the container as unstable or restart.	Mandatory	`5`
`designer.designerSecrets.enabled`	This enables providing the GWS Client ID and Secret as an input to the Designer pods. Kubernetes Secrets is used to store the GWS client credentials.	Mandatory	`true`
`designer.designerSecrets.secrets`	GWS Client ID and GWS Client Secret. Create a new GWS Client if it does not exist. A link to information on creating a new GWS Client is provided in the *Platform settings* section.	Mandatory	
`designer.service.enabled`	Set to `true` if the service must be created.	Optional	`true`
`designer.service.type`	Service type. Valid values are: `ClusterIP`, `NodePort`, `LoadBalancer`.	Mandatory	`NodePort`
`designer.service.port`	The Designer service port to be exposed in the cluster.	Mandatory	`8888`
`designer.service.targetPort`	The Designer application port running inside the container.	Mandatory	`http`
`designer.service.nodePort`	Port to be exposed in case service type is `NodePort`.	Mandatory for `designer.service.type=NodePort`.	`30180`

`designer.service.termination_grace_period`	The period after which Kubernetes starts to delete the pods after service termination.	Optional	`30` seconds.
`designer.ingress.enabled`	Set to `true` to enable ingress. Ingress should be enabled for all cases except for a lab/demo setup.	Mandatory	`true`
`designer.ingress.apiVersion`	The **apiVersion** of the ingress manifest to be deployed. Currently, `networking.k8s.io/v1beta1` and `networking.k8s.io/v1` are supported.	Optional	`networking.k8s.io/v1`
`designer.ingress.ingressClassName`	The ingress class name for the ingress deployed. Applicable only when `designer.ingress.apiVersion` is `networking.k8s.io/v1`.	Optional	
`designer.ingress.annotations`	Annotations added for ingress. The Designer UI requires Session Stickiness if the replica count is more than 1. Configure Session Stickiness based on the ingress controller type. Configuration specific to ingress such as Session Stickiness can be provided here.	Optional	
`designer.ingress.paths`	Ingress path	Mandatory	`[/]`
`designer.ingress.hosts`	Hostnames to be configured in ingress for the Designer service.	Mandatory	- .example.com - .blue.example.com - .green.example.com
`designer.ingress.tls`	TLS configuration for ingress.	Optional	`[]`
`designer.resources.limits.cpu`	Maximum amount of CPU that K8s allocates for the container.	Mandatory	`600m`
`designer.resources.limits.memory`	Maximum amount of memory that K8s allocates for the container.	Mandatory	`1Gi`
`designer.resources.requests.cpu`	Guaranteed CPU allocation for the	Mandatory	`500m`

	container.		
`designer.resources.requests.memory`	Guaranteed memory allocation for the container.	Mandatory	`512Mi`
`designer.securityContext.runAsUser`	This setting controls which user ID the containers are run with. This can be configured to run Designer as a non-root user. You can either use the **Genesys** user or arbitrary UIDs. Both are supported by the Designer base image. `500` is the ID of the Genesys user. The file system must reside within the Genesys user account in order to run Designer as a Genesys user. Change the NFS server host path to the Genesys user: `chown -R genesys:genesys`.	Optional	
`designer.securityContext.runAsGroup`	Controls which primary group ID the containers are run with. This can be configured to run Designer as a non-root user. You can either use the **Genesys** userGroup (GID - `500`) or arbitrary GIDs. Both are supported by the Designer base image.	Optional	

`designer.nodeSelector`	To allow pods to be scheduled based on the labels assigned to the nodes.	Optional	Default value:
			`nodeSelector: {}`
			Sample value:
			`nodeSelector:`
			`    :`

`designer.affinity`	The K8s standard node affinity and anti-affinity configurations can be added here. Refer to the this topic in the Kubernetes documentation site for sample values.	Optional	`{}`
`designer.tolerations`	Tolerations work with taints to ensure that pods are not scheduled on to inappropriate nodes. Refer to the	Optional	`[]`

	Taints and Tolerations topic in the Kubernetes documentation site for sample values.		
`designer.podDisruptionBudget.enabled`	Set to `true` if a pod disruption budget is to be created.	Optional	`false`
`designer.podDisruptionBudget.minAvailable`	The number of pods that should always be available during a disruption.	Optional	`1`
`designer.dnsPolicy`	The DNS policy that should be applied to the Designer pods.	Optional	
`designer.dnsConfig`	The DNS configuration that should be applied to the Designer pods.	Optional	
`designer.priorityClassName`	The priority class name that the pods should belong to.	Optional	
`designer.hpa.enabled`	Enables K8s Horizontal Pod Autoscaler (HPA). It automatically scales the number of pods based on average CPU utilization and average memory utilization. For more information on HPA refer to this topic in the Kubernetes documentation site.	Optional	`false`
`designer.hpa.targetCPUPercent`	The K8s HPA controller will scale up or scale down pods based on the target CPU utilization percentage specified here. It scales up or scales down pods between the range - `designer.deployment.replicaCount` and `designer.deployment.maxreplicaCount`.	Optional	`70`
`designer.hpa.targetMemoryPercent`	The K8s HPA controller will scale up or scale down pods based on the target memory utilization percentage specified here. It scales up or scales down pods between the range - `designer.deployment.replicaCount` and `designer.deployment.maxreplicaCount`.	Optional	`70`

`designer.labels`	Labels that will be added to the Designer pods.	Optional	`{}`
`designer.annotations`	Annotations added to the Designer pods.	Optional	`{}`
`designer.prometheus.enabled`	Set to `true` if Prometheus metrics must be enabled.	Optional	`false`
`designer.prometheus.tagName`	Label key assigned to the pods/service to filter out.	Optional	`service`
`designer.prometheus.tagValue`	Label value assigned to the pods/service to filter out.	Optional	`designer`
`designer.prometheus.instance`		Optional	`{{instance}}`
`designer.prometheus.serviceMonitor.enabled`	Set to `true if a service monitor resource is needed to monitor the pods through the Kubernetes service.`	Optional	`false`
`designer.prometheus.serviceMonitor.path`	The path in which the metrics are exposed.	Optional	`/metrics`
`designer.prometheus.serviceMonitor.interval`	The scrape interval specified for the Prometheus server. That is, the time interval at which the Prometheus server will fetch metrics from the service.	Optional	`10s`
`designer.prometheus.serviceMonitor.labels`	Labels to be specified for the service monitor resource.	Optional	
`designer.prometheus.alerts.enabled`	Set to `true if Prometheus alerts must to be created.`	Optional	`false`
`designer.prometheus.alerts.customalerts`	Any custom alerts that are created must be specified here.	Optional	
`designer.prometheus.alerts.labels`	Labels to be specified for the alerts resource.	Optional	

`designer.prometheus.alerts`	Scenarios for which alerts need to be created.	Optional	**designer.prometheus.alerts**
			`containerRestartAlert:`
			`interval: 3m`
			`threshold: 5`
			`AlertPriority: CRITICAL`
			`MemoryUtilization:`

			interval: 1m threshold: 70 AlertPriority: CRITICAL endpointAvailable: interval: 1m AlertPriority: CRITICAL CPUUtilization: interval: 1m threshold: 70 AlertPriority: CRITICAL containerReadyAlert: interval: 1m readycount: 1 AlertPriority: CRITICAL WorkspaceUtilization: interval: 3m threshold: 80 workspaceClaim: designer-managed-disk AlertPriority: CRITICAL AbsentAlert: interval: 1m AlertPriority: CRITICAL Health: interval: 3m AlertPriority: CRITICAL WorkspaceHealth: interval: 3m AlertPriority: CRITICAL ESHealth: interval: 3m AlertPriority: CRITICAL GWSHealth: interval: 3m AlertPriority: CRITICAL

`designer.grafana.enabled`	Set to `true` if the Grafana dashboard is to be created.	Optional	`true`
`designer.grafana.labels`	Labels that have to be added to the Grafana ConfigMap.	Optional	
`designer.grafana.annotations`	Annotations that have to be added to the Grafana ConfigMap.	Optional	
`annotations`	Enables Kubernetes Annotations and adds it to all the resources that have been created. For more information, refer to the Annotations topic in the Kubernetes documentation site.	Optional	`{}`
`labels`	Any custom labels can be configured here. It is a key and value pair, for example, key:"value". These labels are added to all resources.	Optional	`{}`
`podLabels`	Labels that will be added to all application pods.	Optional	`{}`
`podAnnotations`	Annotations that will be added to all application pods.	Optional	`{}`

Designer ConfigMap settings

The following table provides information on the environment variables and service-level settings stored in the Designer ConfigMap.

Parameter	Description	Mandatory?	Default Value
`designer.designerConfig.create`	This enables providing environment variables as an input to the Designer pods. It uses a ConfigMap to store the environment variables.	Mandatory	`true`
`designer.designerConfig.envs.DES_PORT`	Designer port for container (`"port"` in flowsettings.json). The input should be a string, within double quotes.	Mandatory	`"8888"`
`designer.designerConfig.envs.DES_APPSERVER_HOST`	DAS hostname (`"applicationHost"` in flowsettings.json).	Mandatory	`das`

`designer.designerConfig.envs.DES_APPSERVER_PORT`	DAS port (`"applicationPort"` in flowsettings.json). The input should be a string, within double quotes.	Mandatory	`"80"`
`designer.designerConfig.envs.DES_DEPLOY_URL`	This is normally not changed. It is the relative path to the workspace on DAS. The default value `"/workspaces"` should be used always (`"deployURL"` in flowsettings.json).	Mandatory	`"/workspaces"`
`designer.designerConfig.envs.DES_USE_HTCC`	Set to `"true"` so Designer works with GWS. If set to `"false"`, Designer defaults to a local mode and may be used temporarily if GWS is unavailable (`"usehtcc"` in flowsettings.json). Input should be "true" or "false".	Mandatory	`"false"`
`designer.designerConfig.envs.DES_HTCC_SERVER`	GWS server host (`"htccserver"` in flowsettings.json). For example, "gws.genhtcc.com". The input should be a string, within double quotes.	Mandatory	" "
`designer.designerConfig.envs.DES_HTCC_PORT`	GWS server port (`"htccport"` in flowsettings.json). For example, `"80"`. The input should be a string, within double quotes.	Mandatory	" "
`designer.designerConfig.envs.DES_ENABLE_ANALYTICS`	To enable or disable Designer Analytics (`"enableAnalytics"` in flowsettings.json). Input should be "true" or "false".	Optional	`"false"`
`designer.designerConfig.envs.DES_ES_URL`	Elasticsearch URL (`"esUrl"` in flowsettings.json). For example, "http://es-service:9200". The input should be a string, within double quotes.	Optional	" "
`designer.designerConfig.envs.DES_`	Elasticsearch Server Host Name (`"esServer"` in flowsettings.json). For	Optional	" "

	example, `"es-service"`). The input should be a string, within double quotes.		
`designer.designerConfig.envs.DES_ES_PORT`	Elasticsearch port (`"esPort"` in flowsettings.json). For example, `"9200"`. The input should be a string, within double quotes.	Optional	`""`
`designer.designerConfig.envs.DES_FILE_LOGGING_ENABLED`	Enable file logging. If not enabled, Designer will create only verbose logs. Input should be "true" or "false".	Mandatory	`"false"`
`designer.designerFlowSettings.create`	Set to `true` to include the contents of the flowsettings.yaml file in a separate ConfigMap. Input should be true or false.	Optional	`false`
`designer.designerFlowSettings.envs`	The flowsettings.yaml file should contain these keys, so that the file's contents will be included in the ConfigMap. Refer to the Updating the flowsettings file section in the *Deploy Designer* topic for more information on this.	Optional	`{}`

DAS deployment settings

The following table provides information on the DAS deployment settings. These settings are configured in the **das-values.yaml** file. DAS Deployment Settings

Parameter	Description	Mandatory?	Default Value
`das.deployment.replicaCount`	Number of pods to be created.	Mandatory	`2`
`das.deployment.maxreplicaCount`	The maximum number of replicas to be created. It is recommended to configure this setting if auto-scaling is used.	Optional	`10`
`das.deployment.strategy`	The deployment strategy to follow. This determines which type	Mandatory	`rollingupdate`

	of resources are deployed. Valid values are: `rollingupdate, blue-green, blue-green-ingress, blue-green-service, canary`. • **rollingupdate** - default Kubernetes update strategy where resources will be updated using the rolling upgrade strategy. • **blue-green** - for deploying and upgrading the DAS service using the blue-green strategy. • **blue-green-ingress** - for the blue-green upgrade, this is to create an ingress for the first time. • **blue-green-service** - for the blue-green upgrade, this is to create a service for the first time, and update the service during a service cutover. • **canary** - to deploy canary pods along with the blue-green pods.		
`das.deployment.color`	This is to deploy/ upgrade the DAS service using the blue-green upgrade strategy. Valid values are: `blue, green`.	Mandatory for `blue-green` and `blue-green-service` strategies.	
`das.deployment.type`	Type of Kubernetes controller. Valid values is: `StatefulSet` • **StatefulSet** - if the Designer workspace is stored in a remote cloud storage system, such as	Optional	`StatefulSet`

	Azure Files.		
`das.image.repository`	Docker repository that contains the images for DAS.	Mandatory	
`das.image.tag`	DAS image version.	Mandatory	
`das.image.pullPolicy`	DAS image pull policy (imagePullPolicy). Valid values are: `Always, IfNotPresent, Never`. • **Always** - always pull the image. • **IfNotPresent** - pull the image only if it does not already exist on the node. • **Never** - never pull the image.	Optional	`IfNotPresent`
`das.image.imagePullSecrets`	Secret name containing the credentials for authenticating access to the Docker repository.	Mandatory	

`das.podVolumes`	Provides the name of the volume and name of the persistent volume claim to be attached to the pods	Mandatory	`das:`
			`podVolumes:`
			`- name: workspace`
			`persistentVolumeClaim:`
			`claimName: designer-managed-disk`
			`- name: logs`
			`persistentVolumeClaim:`
			`claimName: designer-logs`

`das.volumes.podPvc.create`	This volume is usually created to mount a local disk to a DAS container for syncing data in case cloud storage is used for storing Designer files. This value has to be `true` or `false` depending on whether the local disk is needed or not	Optional	`false`
`das.volumes.podPvc.mountPath`	The path where the workspace volume is to be mounted inside the DAS container.	Optional	

`das.volumes.podPvc.claim`	Persistent volume claim name for the volume.	Optional	`local-workspace`
`das.volumes.podPvc.claimSize`	Size of the persistent volume claim for the pod. The persistent volume must be equal to or greater than this size.	Optional	
`das.volumes.podPvc.storageClass`	storageClassName provided in the persistent volume that is created for DAS (example, **nfs**).	Optional	
`das.volumes.podPvc.accessModes`	The read/write priveleges and mount priveleges of the volume claim with respect to the nodes. Valid types are: `ReadWriteOnce`, `ReadOnlyMany`, `ReadWriteMany`. • **ReadWriteOnce** - the volume can be mounted as read-write by a single node. • **ReadOnlyMany** - the volume can be mounted as read-only by many nodes. • **ReadWriteMany** - the volume can be mounted as read-write by many nodes. For more information, refer to the access modes topic in the Kubernetes documentation site.	Optional	`ReadWriteOnce`

`das.volumeMounts`	The name of the volume and the mount path to be used by the pods.	Mandatory	`volumeMounts:`
			`- mountPath: /das/www/workspaces`
			`name: workspace`
			`- mountPath: /das/log`
			`name: logs`

`das.dasSecrets.enabled`	Set to `true` if Kubernetes secrets must be created to store	Optional	`false`

	keys/credentials/tokens.		
`das.dasSecrets.secrets`	Key value pairs containing the secret, such as, username and password.	Optional	
`das.livenessProbe.path`	DAS liveness probe API path.	Mandatory	`/health`
`das.livenessProbe.containerPort`	Port running the container.	Mandatory	`8081`
`das.livenessProbe.startupDelay`	The liveness probe will be started after a given delay as specified here.	Mandatory	`10`
`das.livenessProbe.checkInterval`	The interval between each liveness probe request.	Mandatory	`5`
`das.livenessProbe.failureCount`	Number of liveness probe failures after which, to mark the container as unstable or restart.	Mandatory	`3`
`das.readinessProbe.path`	DAS readiness probe API path.	Mandatory	`/health`
`das.readinessProbe.containerPort`	Port running the container.	Mandatory	`8081`
`das.readinessProbe.startupDelay`	The readiness probe will be started after a given delay as specified here.	Mandatory	`10`
`das.readinessProbe.checkInterval`	The interval between each readiness probe request.	Mandatory	`5`
`das.readinessProbe.failureCount`	Number of readiness probe failures after which, to mark the container as unstable or restart.	Mandatory	`3`
`das.service.enabled`	Set to `true` if the service must be created.	Optional	`true`
`das.service.type`	Service type. Valid values are: `ClusterIP`, `NodePort`, `LoadBalancer`.	Mandatory	`NodePort`
`das.service.port`	The DAS service port to be exposed in the cluster.	Mandatory	`80`
`das.service.targetPort`	The DAS application port running inside the container.	Mandatory	`http`
`das.service.nodePort`	Port to be exposed in	Mandatory if	`30280`

	case service type is `NodePort`.	`das.service.type` is `NodePort`.	
`das.service.termination_grace_period`	The period after which Kubernetes starts to delete the pods in case of deletion.	Optional	`30` seconds.
`das.ingress.enabled`	Set to `true` to enable ingress. Ingress should be enabled for all cases except for a lab/demo setup.	Optional	`false`
`das.ingress.apiVersion`	The **apiVersion** of the ingress manifest deployed. Supported versions are, `networking.k8s.io/v1beta1` and `networking.k8s.io/v1`.	Optional	`networking.k8s.io/v1`
`das.ingress.ingressClassName`	The ingress class name for the ingress deployed. Applicable only when `das.ingress.apiVersion` is `networking.k8s.io/v1`.	Optional	
`das.ingress.annotations`	Annotations added for the ingress resources.	Optional	
`das.ingress.paths`	Ingress path.	Optional	`[/]`
`das.ingress.hosts`	Hostnames to be configured in ingress for the DAS service.	Mandatory if ingress is enabled.	
`das.ingress.tls`	TLS configuration for ingress.	Optional	`[]`
`das.resources.limits.cpu`	Maximum amount of CPU that K8s allocates for the container.	Mandatory	600m
`das.resources.limits.memory`	Maximum amount of memory that K8s allocates for the container.	Mandatory	1Gi
`das.resources.requests.cpu`	Guaranteed CPU allocation for the container.	Mandatory	`400m`
`das.resources.requests.memory`	Guaranteed memory allocation for the container.	Mandatory	`512Mi`
`das.securityContext.runAsUser`	This setting controls which user ID the containers are run with and can be configured	Optional	

	to run DAS as a non-root user. You can either use the **Genesys** user or arbitrary UIDs. Both are supported by the DAS base image. `500` is the ID of the Genesys user. For more information refer to the Security Context topic in the Kubernetes documentation site.		
`das.securityContext.runAsGroup`	This setting controls which primary group ID the containers are run with and can be configured to run DAS as a non-root user. You can either use the **Genesys** userGroup (GID - `500`) or arbitrary GIDs. Both are supported by the DAS base image.	Optional	

`das.nodeSelector`	To allow pods to be scheduled based on the labels assigned to the nodes.	Optional	Default value:
			`nodeSelector: {}` Sample value:
			`nodeSelector:`
			`    :`

`das.affinity`	The K8s standard node affinity and anti-affinity configurations can be added here. Refer to the this topic in the Kubernetes documentation site for sample values.	Optional	`{}`
`das.tolerations`	Tolerations work with taints to ensure that pods are not scheduled on to inappropriate nodes. Refer to the Taints and Tolerations topic in the Kubernetes documentation site for sample values.	Optional	`[]`
`das.podDisruptionBudget.enabled`	Set to `true` if a pod disruption budget is to be created.	Optional	`false`
`das.podDisruptionBudget.minAvailable`	The number of pods that should always be available during a disruption.	Optional	`1`

`das.dnsPolicy`	The DNS policy that should be applied to the DAS pods.	Optional	
`das.dnsConfig`	The DNS configuration that should be applied to the DAS pods.	Optional	
`das.priorityClassName`	The priority class name that the pods should belong to.	Optional	
`das.hpa.enabled`	Set to `true` if a K8s Horizontal Pod Autoscaler (HPA) is to be created.	Optional	`false`
`das.hpa.targetCPUPercent`	The K8s HPA controller will scale up/down pods based on the target CPU utilization percentage specified. It scale up/down pods between the range `deployment.replicaCount` to `deployment.maxReplicas`	Optional	75
`das.hpa.targetMemoryPercent`	The K8s HPA controller will scale up or scale down pods based on the target CPU utilization percentage specified here. It scales up or scales down pods between the range - `deployment.replicaCount` and `deployment.maxReplicas`.	Optional	`70`
`das.labels`	Labels that will be added to the DAS pods.	Optional	`{}`
`das.annotations`	Annotations added to the DAS pods.	Optional	`{}`
`das.prometheus.enabled`	Set to `true` if Prometheus metrics must be enabled.	Optional	`false`
`das.prometheus.tagName`	Label key assigned to the pods/service to filter out.	Optional	`service`
`das.prometheus.tagValue`	Label key assigned to the pods/service to filter out.	Optional	`designer`
`das.prometheus.pod`		Optional	`{{pod}}`
`das.prometheus.instance`		Optional	`{{instance}}`
`das.prometheus.serviceMonitor`	Set to `true if a service monitor`	Optional	`false`

	`resource is needed to monitor the pods through the Kubernetes service.`		
`das.prometheus.serviceMonitor.path`	The path in which the metrics are exposed.	Optional	`/metrics`
`das.prometheus.serviceMonitor.interval`	The scrape interval specified for the Prometheus server. That is, the time interval at which the Prometheus server will fetch metrics from the service.	Optional	`10s`
`das.prometheus.serviceMonitor.labels`	Labels to be specified for the service monitor resource.	Optional	
`das.prometheus.alerts.e`	Set to `true if Prometheus alerts must to be created.`	Optional	`false`
`das.prometheus.alerts.labels`	Labels to be specified for the alerts resource.	Optional	
`das.prometheus.alerts.customalerts`	Any custom alerts that are created must be specified here.	Optional	

`das.prometheus.alerts.`	Scenarios for which alerts need to be created.	Optional	**das.prometheus.alerts.**
			`containerRestartAlert:`
			`interval: 3m`
			`threshold: 5`
			`AlertPriority: CRITICAL`
			`MemoryUtilization:`
			`interval: 1m`
			`threshold: 75`
			`AlertPriority: CRITICAL`
			`endpointAvailable:`
			`interval: 1m`
			`AlertPriority: CRITICAL`
			`CPUUtilization:`
			`interval: 1m`
			`threshold: 75`
			`AlertPriority: CRITICAL`
			`containerReadyAlert:`
			`interval: 5m`
			`readycount: 1`

			AlertPriority: CRITICAL rsyncContainerReadyAlert: interval: 5m readycount: 1 AlertPriority: CRITICAL WorkspaceUtilization: interval: 3m threshold: 70 workspaceClaim: designer-managed-disk AlertPriority: CRITICAL AbsentAlert: interval: 1m AlertPriority: CRITICAL LocalWorkspaceUtilization: interval: 3m threshold: 70 AlertPriority: CRITICAL Health: interval: 3m AlertPriority: CRITICAL WorkspaceHealth: interval: 3m AlertPriority: CRITICAL PHPHealth: interval: 3m AlertPriority: CRITICAL ProxyHealth: interval: 3m AlertPriority: CRITICAL PhpLatency: interval: 1m threshold: 10 AlertPriority: CRITICAL HTTPLatency: interval: 1m

			`threshold: 60`
			`AlertPriority: CRITICAL`
			`HTTP4XXCount:`
			`interval: 5m`
			`threshold: 100`
			`AlertPriority: CRITICAL`
			`HTTP5XXCount:`
			`interval: 5m`
			`threshold: 100`
			`AlertPriority: CRITICAL`

das.grafana.enabled	Set to `true` if the Grafana dashboard is to be created.	Optional	`true`
das.grafana.labels	Labels that must be added to the Grafana ConfigMap.	Optional	
das.grafana.annotations	Annotations that must be added to the Grafana ConfigMap.	Optional	
`annotations`	Enables Kubernetes Annotations and adds it to all the resources that have been created. For more information, refer to the Annotations topic in the Kubernetes documentation site.	Optional	`{}`
`labels`	Any custom labels can be configured here. It is a key and value pair, for example, key:"value". These labels are added to all resources.	Optional	`{}`
`podLabels`	Labels that will be added to all application pods.	Optional	`{}`
`podAnnotations`	Annotations that will be added to all application pods.	Optional	`{}`

DAS ConfigMap settings

Parameter	Description	Mandatory?	Default Value

`das.dasConfig.create`	This setting enables providing environment variables as an input to the DAS pods. It uses a ConfigMap to store the environment variables.	Mandatory	`true`
`das.dasConfig.envs.DAS_FILE_LOGGING_ENABLED`	Enables file logging. DAS supports only std out logging. This should always be set to `false`. Input should be "true" or "false".	Mandatory	`"false`
`das.dasConfig.envs.DAS_`	Enables log levels. Valid values are: `"FATAL"`, `"ERROR"`, `"WARN"`, `"INFO"`, `"DEBUG"`, `"TRACE"`.	Optional	`"DEBUG"`
`das.dasConfig.envs.DAS_STDOUT_LOGGING_ENABLE`	Enables standard output console logging. Input should be "true" or "false".	Mandatory	`"true"`
`das.dasConfig.envs.DAS_SERVICES_ELASTICSEARCH_ENABLED`	To enable Designer Analytics. This configuration is required for DAS to initialize ES templates. Input should be "true" or "false".	Optional	`"false"`
`das.dasConfig.envs.DAS_SERVICES_ELASTICSEARCH_HOST`	Elasticsearch server host name with an http:// prefix. For example, http://es-service. The input should be a string within double quotes.	Optional	" "
`das.dasConfig.envs.DAS_SERVICES_ELASTICSEARCH_PORT`	Elasticsearch port. For example, `"80"`. The input should be a string, within double quotes.	Optional	" "
das.dasConfig.envs.DAS_ELASTIC_URL	Elasticsearch URL for basic authentication. It should contain the URL with an http or https prefix accompanied with the port number (for example, http://es-service:80). The input should be a string within double quotes. This setting is mandatory when `DAS_SERVICES_ELASTICSEARCH_ENABLED` is set to `true`.	Optional	" "
das.dasConfig.envs.DAS_ELASTIC_URL_	Elasticsearch secondary	Optional	" "

	region URL for basic authentication. It should contain the URL with an http or https prefix accompanied with the port number (for example, http://es-service:80). The input should be a string within double quotes. is an integer starting from 1. This setting is mandatory when secondary regions are configured. For example, `das.dasConfig.envs.DAS_ELASTIC_URL_1`.		

Post deployment Designer configuration settings

Post deployment, Designer configuration is managed from the following 3 locations:

Flow settings

Flow Settings is used for controlling global Designer settings that are applicable to all tenants and it contains bootstrap configuration settings such as port, GWS info, and DAS URL.

Configuration path - `/workspace/designer/flowsettings.json`.

This will be configured using the helm install. Refer to the Update the flowsettings.json file section for information on updating the **flowsettings.json** file.

Tenant settings

These are tenant specific settings if the Designer service is configured with multi-tenancy .

Configuration path - `workspace//config/tenantsettings.json`.

The user should logout and log back in after any changes to the **tenantsettings.json** file. The Designer UI will continue to show the older features until the user logs out and logs back in.

Tenant specific settings are configured by directly editing the file in the above path.

DesignerEnv transaction list

The **DesignerEnv** transaction list is available in Configuration Server (`Tenant/Transactions/DesignerEnv`). This is mostly used to control the run-time settings. Any change to the **DesignerEnv** transaction list does not require the application to be published again or a new build for the application.

The user should log out and log back in for the changes to reflect in the Designer UI.

The **DesignerEnv** transaction list is configured using Agent Setup.

Post deployment configuration settings reference table

Category: Analytics					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`enableAnalytics` (optional)	Yes	Yes	No	This flag enables or disables the analytics feature.	Sample value: `true` Default value: `false`
`esUrl` (optional)	Yes	Yes	No	Elasticsearch URL	Sample value: http://es-spot.usw1.genhtcc.com:80
`esServer` (optional)	Yes	Yes	No	Elasticsearch server host name (for example, `es-service`).	Sample value: es-spot.usw1.genhtcc.com
`esPort` (optional)	Yes	Yes	No	Elasticsearch port.	Sample value: `80`
`ReportingURL` (optional)	No	No	Yes Section: `reporting`	URL of Elasticsearch where Designer applications will report data.	Sample value: http://es-spot.usw1.genhtcc.com:80
`esMaxQueryDuration` (optional)	Yes	Yes	No	The maximum time range (in days) to query in Designer Analytics. Each day's data is stored in a separate index in Elasticsearch.	Sample value: `90` Default value: `90`
`sdrMaxObjCount` (optional)	Yes	Yes	No	The maximum count of nested type objects that will be captured in SDRs. When set to -1, which is the default value, no objects will be trimmed. All	Sample value: `20`

				the *milestones* or *activities* visited in runtime are expected to be captured in an SDR.	
`SdrTraceLevel` (optional)	Yes	Yes	No	Value are: • **100** — Debug level and up. Currently, there are no Debug messages. • **200** — Standard level and up. This setting will show all blocks that are entered during a call in the `blocks` array. • **300** — Important level and up. This setting filters out all blocks from the `blocks` array, except those containing data that will change from call to call (such as the Menu block and User Input block).	Sample value: `300` Default value: `300`
Category: Audit					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`enableESAuditLogs` (optional)	Yes	Yes	No	Enable or disable audit logs captured in Elasticsearch.	Sample value: `false` Default value: `false`
`enableFSAuditLogs` (optional)	Yes	Yes	No	Enable or Disable audit logs captured in the file system under the **logs** directory or in standard output.	Sample value: `true` Default value: `true`
`maxAppSizeCompare` (optional)	Yes	Yes	No	The maximum size of data object for which a difference will be captured in the audit logs, value in bytes. That is, the difference between the Designer object's old value and new value.	Sample value: `1000000` Default value: `1000000`
`enableReadAuditLogs` (optional)	Yes	Yes	No	Control whether reading of Designer objects is captured in audit trails. If enabled any Designer object viewed in the UI will be recorded in the audit logs.	Sample value: `false` Default value: `false`
Category: Authorization					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`disableRBAC` (optional)	Yes	Yes	No	Controls if Designer reads and enforces permissions associated with the logged in user's roles.	Sample value: `false` Default value: `false`
`rbacSection` (optional)	Yes	Yes	No	In a Role object, the name of the	Sample value: `CfgGenesysAdministratorServer`

				section within the Annex where the privileges are stored.	Default value: `CfgGenesysAdministratorServer`
`disablePBAC` (optional)	Yes	Yes	No	Controls if Designer allows partitioning of the Designer workspace and restricts a user's access to Designer objects in the user's partitions.	Sample value: `false` Default value: `false`
Category: Collaboration					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`locking` (optional)	Yes	No	No	The type of locking used, in an editing session for applications, modules, or data tables. Valid values are: `file, redis, none`. - **none** - resources are not locked and can be edited simultaneously by multiple users which can result in one user overwriting another user's changes. - **file** - uses files to keep track of locks and relies on shared storage (for example, NFS) to	Sample value: `file` Default value: `file`

				make lock files available to each Designer pod. Lock files are stored in the same location as the user's Designer workspace. • **redis** - uses Redis for storing resource locks and is recommended for production environments.	
Category: DAS					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`applicationHost` (mandatory)	Yes	No	No	The server name Designer uses to generate the URL to the application. ORS and MCP fetch the application code and other resources from this URL.	Sample value: `das.usw1.genhtcc.com` Default value: `localhost`
`applicationPort`	Yes	No	No	The corresponding port to be used with applicationHost.	Sample value: `80` Default value: `80`
`deployURL`	Yes	No	No	This is normally not changed. It is the relative path to the workspace on DAS.	Sample value: `/workspace` Default value: `/workspace`
Category: Digital					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`rootsSRL` (optional)	Yes	Yes	No	If specified, this is used to filter which Root Categories to display when selecting Standard Responses.	Sample value: Any REGular EXpression (REGEX).
`maxFlowEntryCount` (optional)	Yes	No	Yes Section: `flowsettings`	Specify how many times the same application can process a specific digital interaction.	Sample value: `20` Default value: `20`
Category: External APIs					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`httpProxy` (optional)	Yes	Yes	Yes Secion: `flowsettings`	Specify the proxy used for external requests and nexus API calls (if `enable_proxy` is `true`).	Sample value: [http://vpcproxy-000-int.geo.genprim.
`redundantHttpProxy` (optional)	Yes	Yes	Yes Section: `flowsettings`	Specify the backup proxy used for external requests and nexus API calls (if `enable_proxy` is `true`), when httpProxy is down.	Sample value: [http://vpcproxy-001-int.geo.genprim.
Category: Features					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`features`	Yes	Yes	No	This is an object. See the **5.5 Features** section for a list of supported features.	Default value:
					`{`
					`nexus: true,`
					`enableBulkAudioImport: true`
					`}`

Category: GWS					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`usehtcc`	Yes	No	No	Set to `true` so that Designer works with GWS. If set to `false`, Designer defaults to a local mode and may be used temporarily if GWS is unavailable.	Sample value: `true` Default value: `false`
`htccServer`	Yes	No	No	GWS Server	Sample value: gws-usw1-int.genhtcc.com Default value: gws-usw1-int.genhtcc.com
`htccport`	Yes	No	No	GWS port.	Sample value: `80` Default value: `80`
`ssoLoginUrl`	Yes	No	No	URL of GWS authentication UI. Designer redirects to this URL for authentication.	Sample value: https://gws-usw1.genhtcc.com Default value: https://gws-usw1.genhtcc.com
`maxConcurrentHTCCRequest` (optional)	Yes	No	No	For batch operations to GWS, the max number of concurrent requests that Designer will send to GWS.	Sample value: `5` Default value: `5`
`batchOperationResultTTL` (optional)	Yes	No	No	For batch operations to GWS, the time, in milliseconds, for which duration Designer stores the results of a batch operation on the server, before deleting them.	Sample value: `100000` Default value: `100000`
Category: Help					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`docsMicroserviceURL` (optional)	Yes	No	No	URL for Designer documentation.	Default value: https://docs.genesys.com/Documentation/

					PSAAS/Public/ Administrator/ Designer
Category: IVR					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`recordingType` (optional)	Yes	Yes	No	Specify the recording type to be used in Record block. Set as `GIR`. If the option is missing or blank, `Full Call Recording` type will be used.	Sample value: `GIR` Default value: `GIR`
Category: Logging					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`logging: {`
`designer: {`
`level:`
`debug },`
`audit: {`
`level:`
`trace},`
`auditdebug:`
`{ level:`
`debug },`
`cli: {`
`level: debug`
`}`
`}`
(optional)
	Yes	No	No	Specify Designer log levels. Each field has valid values: `trace, debug, info, warn, error, or fatal`.
				• **designer** - log level of Designer.
				• **audit** - log level of audit.
				• **auditdebug** - log level of audit debug, this will log detailed audit information.
				• **cli** - log level for cli commands executed on Designer.
					Sample value:
					`logging: {`
					`designer: {`
					`level:`
					`debug},`
					`audit: {`
					`level: trace`
					`},`
					`auditdebug:`
					`{ level:`
					`debug},`
					`cli: {`
					`level: debug`
					`}`
					`}`
					Default value:
					`logging: {`
					`designer: {`
					`level: debug`
					`},`
					`audit: {`
					`level: trace`
					`},`
					`auditdebug:`
					`{ level:`
					`debug },`
					`cli: {`
					`level: debug`
					`}`
					`}`

Category: Nexus					

Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
url (optional)	No	No	Yes Section: nexus	URL of Nexus that typically includes the API version path. For example, https://nexus-server/nexus/api/v3.	Default value: http://nex-dev.usw1.genhtcc.com
password (optional)	No	No	Yes Section: nexus	The Nexus x-api-key created by Nexus deployment.	Default value: dc4qeiro13nsof569dfn234smf
enable_proxy (optional)	No	No	Yes Section: nexus	Boolean value to indicate if httpProxy is used to reach Nexus. Default value: false	
profile (optional)	No	No	Yes Section: nexus	Enable Contact Identification via Nexus (for example, to enable Last Called Agent routing).	
Category: Process					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
port	Yes	No	No	Designer process port in the container. Normally, the default value should be left as is.	Sample value: 8888 Defualt value: 3000
Category: Provisioning					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
primarySwitch (optional)	Yes	Yes	No	Specify the primary switch name if more than one switch is defined for the tenant. Designer fetches and works with route points from this	Default value: us-west-1

				switch.	
Category: Routing					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`ewtRefreshTimeout` (optional)	NO	No	Yes Section: `flowsettings`	Specify the interval (in seconds) at which to refresh the Estimated Waiting Time when routing an interaction.	Sample value: 5 Default value: 1
Category: Redis					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value

`redis: {`
`host: "",`
`port: "",`
`tlsEnabled: true,`
`lockTimeout: 120,`
`listTimeout: 1800`
`}`
(optional)
	Yes	No	No	Used by Designer for resource index caching and multi-user collaboration locks on Designer resources.
				It is a separate object that contains:
				• **host** - Redis host name.
				• **port** - Redis port.
				• **tlsEnabled** - TLS enabled or not.
				• **lockTimeout** - Timeout, in seconds, before a resource lock is released for an editing session of applications, modules, or data tables.
				• **listTimeout** - The cache expiry timeout (in
					Sample value:
					`redis: {`
					`host: "",`
					`port: "",`
					`tlsEnabled: true,`
					`lockTimeout: 120,`
					`listTimeout: 1800`
					`}`
					Default value:
					`redis: {`
					`host: redis.server.genhtcc.com,`
					`port: 6379,`
					`tlsEnabled: true,`
					`lockTimeout: 120,`
					`listTimeout: 1800`
					`}`

				seconds) of the application list and shared modules list. By default, it is 30 minutes. That is, any new application/ modules created in the UI will be seen in the listing page after 30 mins. It can be reduced to a smaller value. This is to improve the page loading performance of the Applications and Shared Modules page. A better performance is achieved with a higher value.	
Category: Security					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`zipFileSizeLimitInMegaBytes` (optional)	Yes	Yes	No	Defines the maximum zipFile size limit (in megabytes) during bulk audio import.	Sample value: `50`
`disableCSRF` (optional)	Yes	Yes	No	Disable CSRF attack protection. For more information, refer to this	Sample value: `false` Default value: `false`

				topic in the CWE site. By default, CSRF attack protection is enabled. It can be disabled by setting this flag to `true`.	
`disableSecureCookie` (optional)	Yes	No	No	Disables the secure cookies header.	Sample value: `false` Default value: `false`
Category: Session					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`idleTimeout` (optional)	Yes	Yes	No	Idle timeout, in seconds, before a user session is terminated while editing applications, modules, or data tables.	Sample value: `840` Default value: `840`
`lockTimeout` (optional)	Yes	Yes	No	Timeout, in seconds, before a resource lock is released, for an editing session of applications, modules, or data tables.	Sample value: `120` Default value: `120`
`lockKeepalive` (optional)	Yes	Yes	No	Interval, in seconds, before the client sends a ping to the server, to refresh the lock for an editing session of applications, modules, or data tables.	Sample value: `15` Default value: `15`
Category: Workflow					
Setting Name	flowsettings.json	tenantsettings.json	DesignerEnv	Description	Value
`maxBuilds` (optional)	Yes	Yes	No	Specify the maximum number of builds allowed per application.	Sample value: `20` Default value: `20`

enablePTE (optional)	No	No	Yes Section: flowsettings	Boolean value to indicate if PTE objects are enabled at runtime.	Sample value: true Default value: false

Features

The features specified in this section are configured under the `features` object in the **flowsettings.json** file or the **tenantsettings.json** file.

For example,

```
"features": {
            "nexus": true,
..
}
```

> **Important**
>
> These features are configured only in the **flowsettings.json** file and the **tenantsettings.json** file, and not in the **DesignerEnv** transaction list.

Category	Feature Setting Name	Mandatory	flowsettings.json	tenantsettings.json	Description	Default Value
Audio	enableBulkAudioImport	Optional	Yes	Yes	Enable/ disable the bulk audio import feature.	false
	grammarValidation	Optional	Yes	yes	If this feature is enabled, Designer will validate invalid grammar files during grammar upload and you can upload only valid grammar files (GRXML or Nuance compiled binary grammar files).	false

	`externalAudioSupport`	Optional	Yes	Yes	If this feature is enabled, a new audio type, External Audio, is available in the Play Message block. It accepts a single variable that contains a URL to the audio resource. MCP will fetch this resource directly and play it. The only supported value of **Play As** is *Audio URI*. There is no automatic language switching for this audio type.	`false`
Nexus	`nexus`	Optional	Yes	Yes	Enable/ disable the Nexus feature.	`false`
Survey	`survey`	Optional	Yes	Yes	Enable/ disable the survey feature.	`true`
UI Plugins	`plugins`	Optional	Yes	Yes	Plugin configuration details. (Steps are given below the table.)	`{}`
	`plugins`	Optional	Yes	Yes	Enable or disable the plugin feature.	`false`
Milestone	`enableImplicitModuleMilestones`	Optional	Yes	Yes	Enable reporting	`false`

					each Shared Module call as an internal milestone. If disabled, Shared Module calls will not generate a milestone.	
Bots	`enableDialogFlowCXBot`	Optional	Yes	Yes	When enabled, Dialogflow CX bot type is added to the bot registry and available for selection in the **Bot provider** drop-down when you configure a new bot.	`false`
Multisite Routing	`multisiteRouting`	Optional	Yes	Yes	Enables the **Override DN** option in the **Advanced** > **Targeting** section of the **Route Call** block to Force Route the interaction to a specified DN.	`false`

Adding a UI plugin to Designer

1. Add the `plugins` array object in the **flowsettings.json** file (*/ofs/designer/flowsettings.json*).
   The `plugins` object contains all the input properties for the plugin app. This is a required property. Whenever there is a change in this object, refresh the browser for the changes to take effect.
   Example:

```
"plugins": [
     {
       "url": "http://genesysexample.com/",
       "displayName": "Nexus PII Management",
       "placement": "messageCollections",
       "id":   "nexuspii",
       "mappings": {
             "prod": {
                 "G1-AUS4": "https://genesysexample.com/admin/ux"
             },
             "staging": {
                 "G1-USW1": "http://genesysexample.com/"
             },
         }
     },
     {
        ...
     }]
```

2. Add the `csplist` array object in the **flowsettings.json** file (*/ofs/designer/flowsettings.json*).
   The `cspList` object contains the URL forms to be allowed by Designer's security policy. This is a required property. Whenever there is a change in this object, re-start the node container for the changes to take effect.
   Example:
   If the URL is `http://genesysexample.com/`, the `cspList` would be:
   `"cspList": ["*.genexample1.com:*", "*.genexample2.com:*", "*.genexample3.com:*"]`

3. Turn on the `plugins` and `nexus` feature flags in the Designer **tenantSettings.json** file (*/ofs//config/tenantSettings.json*).
   This is a required property. Whenever there is a change in this object, log out of Designer and log in again for the changes to take effect.

   > ### Important
   > If you want to enable the plugins feature for all tenants, add this feature flag in the **flowsettings.json** file. The feature is enabled for all the tenants under that bucket.

   Example:

```
{
    "features": {
        "plugins": true,
        "nexus": true
}}
```

4. Add the `url_` property under the `plugins` section, in Agent Setup. If there is no `plugins` section, create one. This section is for the tenant URL override. If the `DesignerEnv` setting (*Transactions/Internal/DesignerEnv*) is not provided, the plugin URL from the **flowsettings.json** file is considered.
   This is an optional property. Whenever there is a change in this object, log out of Designer and log in again for the changes to take effect.
   Example:
   `{"url_" : "https://plugin-genesysexample.com"}`