Australian Government

Australian Digital Health Agency

HIPS
Upgrade Instructions

17 April 2020 7.3

Approved for external use

Australian Digital Health Agency ABN 84 425 496 912, Level 25, 175 Liverpool Street, Sydney, NSW 2000
Telephone 1300 901 001 or email help@digitalhealth.gov.au
www.digitalhealth.gov.au

Acknowledgements

Council of Australian Governments
The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

HL7 International
This document includes excerpts of HL7TM International standards and other HL7 International material. HL7 International is the publisher and holder of copyright in the excerpts. The publication, reproduction and use of such excerpts is governed by the HL7 IP Policy and the HL7 International License Agreement. HL7 and CDA are trademarks of Health Level Seven International and are registered with the United States Patent and Trademark Office.

Disclaimer
The Australian Digital Health Agency ("the Agency") makes the information and other material ("Information") in this document available in good faith but without any representation or warranty as to its accuracy or completeness. The Agency cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document control
This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

OFFICIAL

Document information

Key information

Owner	National Health Chief Information Officer, Infrastructure Operations	
Contact for enquiries	Australian Digital Health Agency Help Centre	
	Phone	1300 901 001
	Email	help@digitalhealth.gov.au

Table of contents

1 Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Assumptions 5

1.4 Definitions and acronyms 5

2 Upgrade Procedure 6

2.1 Approach 6

2.2 Preparation 6

2.3 Upgrading HIPS Core 7

2.3.1 Remove the previous HIPS-Core version 7

2.3.2 Upgrade the HIPS-Core database 8

2.3.3 Configure & Install the new HIPS-Core Version 10

2.3.4 Adjust Integrated Systems 10

2.3.5 Post-Verification Steps 11

2.4 Upgrading HIPS UI 11

2.4.1 HTTPS & HTTP/2 11

2.4.2 Remove the Previous HIPS-UI Version 11

2.4.3 Upgrade the HIPS-UI Database 12

2.4.4 Configure & Install the new HIPS-UI Version 13

2.5 ESB changes 14

2.6 Finishing 14

3 Troubleshooting 15

3.1 HIPS Core web services 15

3.2 HIPS Core background processes 15

3.3 HIPS UI web application 15

4 Rollback 16

Appendix A Additional Upgrade Notes 17

1 Introduction

1.1 Purpose

This document provides guidance for implementers on upgrading earlier versions of HIPS to HIPS 7.3.

1.2 Scope

The guidance provided in this document is intended for implementers at sites where a version of HIPS v4.2 or later is currently installed and will be upgraded in-place to HIPS 7.3.

For a clean installation of HIPS 7.3 refer to the following documentation provided in the HIPS 7.3 release package:

- *HIPS – Initial and Clean Installation Guide (Core)*
- *HIPS – Initial and Clean Installation Guide (UI)*

1.3 Assumptions

The following assumptions have been made in the development of this document:

- Nil

1.4 Definitions and acronyms

Item	Definition
-	-

2 Upgrade Procedure

2.1 Approach

Earlier versions of HIPS utilised Microsoft Message Queuing (MSMQ) for queuing of operations to be performed in the background. HIPS 7.0.0 onward replaces MSMQ with its own queuing mechanism implemented entirely within the HIPS Core database. The upgrade procedure for HIPS 7.3 caters for the significant changes required to replace MSMQ. For further information on the changes introduced as part of the HIPS 7.0.0 replacement of MSMQ and potential impacts, refer to *Appendix A.1 MSMQ Replacement* in this document.

For upgrade and cutover of existing HIPS installations prior to HIPS 7.0.0 to HIPS 7.3 it is recommended to employ an approach of **draining and in-place upgrade**. Using this approach, systems that are integrated with the previous HIPS installation such as ESBs are configured to queue messages internally for some period. This enables the previous HIPS installation to finalise its processing of “in flight” messages stored in MSMQ. When all processing is complete, the previous HIPS installation is upgraded in-place to HIPS 7.3. Once the upgrade is complete, systems that were configured to queue messages are allowed to interact with the new HIPS installation once more.

It is **strongly recommended** that the upgrade procedure itself be tested first in a non-Production environment.

2.2 Preparation

Perform the following steps to prepare for upgrade to HIPS 7.3:

1 Configure integrated systems such as ESBs to queue messages internally. This is required to prevent them from sending HL7 messages and clinical document uploads to HIPS during the upgrade.

2 If present, consider configuring any load balancer forward of the HIPS deployment to display a service unavailable message for users of HIPS Web UI.

3 Monitor the existing HIPS deployment for processing of “in flight” messages.

a For **versions of HIPS prior to HIPS 7.0**, monitor the status of records in the PcehrMessageQueue table in the HIPS Core database. **Upgrade cannot proceed** until there are **no records** in this table with a QueueStatusId value of 1, indicating that processing of the message is “pending” (or in progress). The following T-SQL script can be executed against the HIPS Core database to assist with monitoring this:

```
SELECT * FROM [hips].[PcehrMessageQueue] WHERE [QueueStatusId] = 1
```

b For **versions of HIPS starting with HIPS 7.0**, monitor the status of records in the MessageQueue table in the HIPS Core database. Upgrade cannot proceed until there are no records in this table with a MessageQueueStateID value of 1, indicating that processing of the message is "pending" (or in progress). The following T-SQL script can be executed against the HIPS Core database to assist with monitoring this:

```
SELECT * FROM [hips].[MessageQueue] WHERE [MessageQueueStateID] = 1
```

Upgrade must not proceed until the applicable query returns 0 results.

4 After all “in flight” messages have been processed, backup the HIPS Core database. This will provide a rollback option in case the upgrade fails and will also provide an archive of queued messages from the existing HIPS deployment.

5 Backup the HIPS runtime folders on each Application Server and the HIPS databases on the Database Server.

6 Stop the existing HIPS web applications in IIS:

a Stop the HIPS-Core and HIPS-UI websites in IIS.

b Stop the application pool(s) used by the HIPS-Core and HIPS-UI websites in IIS.

2.3 Upgrading HIPS Core

2.3.1 Remove the previous HIPS-Core version

HIPS version 7.0.0 onwards largely automates installation and removal of HIPS components via Windows PowerShell. For versions of HIPS earlier than version 7.0.0 that may be installed, it is still necessary however to manually remove HIPS components from Application Servers prior to upgrade.

IMPORTANT: Do not remove the previously created backup of the HIPS runtime folders created on each Application Server. These will be referenced in a subsequent step.

The steps required to remove a previously installed HIPS version from an Application Server depend on the version of HIPS previously installed.

To remove a version earlier than HIPS 7.0

1 Delete HIPS-Core components from IIS:

a Delete the HIPS-Core web application from the HIPS-Core website.

b Delete the HIPS-Core website.

c Delete the application pool used by the HIPS-Core website.

2 Delete HIPS-Core components from the filesystem:

a Browse to the filesystem location referenced by the previously deleted HIPS-Core website and web application.

b Delete the folder and its contents.

IMPORTANT: Do not delete HIPS-Core queues from MSMQ at this time. These are retained in case of the need to rollback, and will be removed in a later step following verification of the upgrade.

To remove HIPS version 7.0 onwards

1. Open a Windows PowerShell console as Administrator.
2. Execute the following command to change directory to the location where the HIPS-Core installation artefacts **for the previously installed HIPS version** are located:
   a. cd <INSTALL_SOURCE_PREVIOUS_VERSION>\HIPS-AppServer\setup
3. Execute the following command to remove all HIPS-Core runtime components using the previously configured installation artefacts:

```
.\Deploy.ps1 -Interactive -ConfigurationDataFile
'Configuration.psd1' -Remove -Components
'SharedConfiguration,Web,QueueConsumer,AlertMonitoring'
```

   a. This command will invoke the Deploy.ps1 PowerShell script using the previously configured Configuration.psd1 file as input to remove each HIPS-Core component if already present.

   b. **NOTE:** If only specific components are installed on a certain Application Server the preceding command can be adjusted accordingly.

4. Ensure the command completes successfully.

2.3.2 Upgrade the HIPS-Core database

IMPORTANT:

In a High Availability topology that employs a SQL Server cluster the HIPS-Core database will be synchronised between cluster nodes.

If filesystem paths differ between cluster nodes the database files created by HIPS-Core database upgrade process may fail to be synchronised. It is recommended to defer synchronisation of the HIPS-Core database to any secondary nodes until after all steps in this section have been completed. If this is not possible, it may be necessary to temporarily remove the HIPS-Core database from synchronisation, complete the upgrade, and then re-initialise synchronisation between cluster nodes.

Locate the folder `<INSTALL_SOURCE>\HIPS-AppServer\`database and application named HIPS.PcehrDataStore.DBUpgrade.exe. This application will install or upgrade the HIPS Core database objects and data into an existing HIPS Core database.

IMPORTANT

Ideally the login used to connect to the specified SQL Server instance must be a member of the *sysadm* fixed server role. If using integrated security, this will be the domain account of the user executing the preceding command. Alternatively, modify the connection string in the preceding command to specify the user name and password for a SQL login with the appropriate membership.

In addition, ensure the default schema of the login's user in the HIPS Core database is set to [dbo]. It should not be set to [hips].

If the [SchemaVersions] table exists in the target database in a schema other than the [dbo] schema, it must be moved to the [dbo] schema prior to execution.

a. Open a command-prompt as Administrator and change directory to the directory which contains the `HIPS.PcehrDataStore.DBUpgrade.exe` application.

b. Execute the following command[1]:

```
.\HIPS.PcehrDataStore.DBUpgrade.exe upgrade-db "Data
Source=#{HIPS.Core.Database.Server};Initial
Catalog=#{HIPS.Core.Database.Name};Integrated Security=SSPI;Connect
Timeout=15;"
```

Modifying the highlighted values as below:

   i. #{HIPS.Core.Database.Server} is the name of the SQL Server instance hosting the database.

[1] Optionally, for help with additional command-line options, execute the following:
`.\HIPS.PcehrDataStore.DBUpgrade.exe upgrade-db --help`

ii. #{HIPS.Core.Database.Name} is the name of the previously-created database to be upgraded.

The command uses a default command timeout of 3600 seconds (1 hour) that can be adjusted via the --timeout option if desired.

If the command was successful, a green ‘Success!’ notice should be displayed. Any errors will be displayed in red.

2. Once the database upgrade is complete, existing audit records from the *IhiLookupAudit*, *HpiiLookupAudit* and *PcehrAudit* tables can be migrated into the new *SystemInteractionLog* table:

IMPORTANT NOTES:

- The migration can be run with configurable batch sizes. Each time the migration is executed it identifies the specified batch size number of records that have not already been migrated and migrates those records. Depending on the batch size and the volume of data to be migrated, the migration may need to be executed multiple times until there are no more batches of data remaining to be migrated.
- It is important that prior to upgrade, the number of audit records are reviewed, and batch sizes planned appropriately. If there is a large volume of data to be migrated, it may be beneficial to execute the migration using smaller batches in one or more suitable time windows where there is less activity in the HIPS database.
- Implementers can optionally drop the [Request] & [Response] columns from the specified audit tables once the entire migration is complete by specifying the --drop-columns argument while executing the migration. The columns will not be dropped if unmigrated data remains after processing a batch.
- Implementers should ensure their database servers have enough storage capacity for database data and log files prior to executing migration.

IMPORTANT NOTE FOR IMPLEMENTERS UPGRADING FROM HIPS 7.0.n to HIPS 7.1+:

- Additional changes have been made to the way that audit data is captured in HIPS 7.1+. This requires implementers to re-execute the audit data migration for HIPS 7.1+. The migration will migrate data that was previously migrated in HIPS 7.0.n to the HIPS 7.1+ structure.

a. Open a command-prompt and change directory to the directory which contains the *HIPS.PcehrDataStore.DBUpgrade.exe* application.

b. Execute the following command:

```
HIPS.PcehrDataStore.DBUpgrade.exe migrate-auditing "Data
Source=#{HIPS.Core.Database.Server};Initial
Catalog=#{HIPS.Core.Database.Name};Integrated Security=SSPI;Connect
Timeout=15;" --audit-type:All --batch-size:100000
```

Modifying the highlighted values as below:

i. #{HIPS.Core.Database.Server} is the name of the SQL Server instance hosting the database.

ii. #{HIPS.Core.Database.Name} is the name of the previously-created database to be upgraded.

iii. The default batch size is 100000. This can be modified, however note that a large batch size will take longer and any error will roll back the migration.

For more help with the migrate auditing utility you can execute
`HIPS.PcehrDataStore.DBUpgrade.exe migrate-auditing -h`

IMPORTANT: Ideally the login used to connect to the specified SQL Server instance must be a member of the *sysadm* fixed server role. If using integrated security, this will be the domain account of the user executing the preceding command. Alternatively, modify the connection string in the preceding command to specify the user name and password for a SQL login with the appropriate membership.

c. If the command was successful, a green 'Success!' notice should be displayed. Any errors will be displayed in red and the transaction will be rolled back.

3. In the same folder as the *HIPS.PcehrDataStore.DBUpgrade.exe* application there is a *ConfigurationScripts* subfolder containing the following script files that can be manually edited and executed if required.

    a. Optional. Use the *Optional_HIPS_Update_Mrn_Padding.sql* script to modify the MRN padding for data already in the HIPS database:

        i. Modify the value of the @length parameter to a value between 0 and 40.

        ii. Execute the script if necessary and ensure it completes successfully.

4. Assign the <HIPS_CORE_SERVICE_ACCOUNT> the VIEW SERVER STATE permission. This is required to run the new Alert Monitoring DBDiskSpace alert processor.

5. Optional. It is possible to reduce the amount of data stored when auditing interactions with the HI Service and My Health Record System by adjusting the value of the [IsEnabled] column of each record in the [hips].[SystemInteractionLogConfiguration] table. For **Non-Production** environments it may be desirable to leave the value of the [IsEnabled] column set to the default value of 1 (Enabled) to assist with diagnostic troubleshooting. For **Production** environments it may be desirable to set the value of the [IsEnabled] column to 0 (Disabled) to limit the amount of data stored for each interaction to meet the minimum conformance requirements.

2.3.3 Configure & Install the new HIPS-Core Version

Perform the following steps on each Application Server, referencing the applicable section from the *HIPS – Initial and Clean Installation Guide (Core)* document:

1. Perform **all steps** from section *Configure Installation Artefacts*. Refer to the configuration files in the HIPS runtime folders previously backed up on each Application Server when configuring the HIPS installation artefacts.
2. Perform **all steps** from section *Execute Installation Scripts* from the *HIPS – Initial and Clean Installation Guide (Core)* document. This will install HIPS-Core runtime components on the Application Server.
3. Perform any applicable additional steps as listed in section *Perform Optional Configuration*.

2.3.4 Adjust Integrated Systems

HIPS 7.0.0 onward simplifies the URLs used to access the HIPS-Core web services by removing the IIS web application previously utilised. This impacts integrated systems that interact with the HIPS-Core web services. Integrated systems must be modified to use the simplified URL scheme.

For example, if the HIPS-Core web services were previously installed into an IIS web application named “HIPS_TEST” within an IIS website exposing HTTP port 12345 on an Application Server named “hips-t-app-01”, they may have been accessed via the following URL:

http://hips-t-app-01:12345/HIPS_TEST/...

This URL must be modified to remove the reference to the HIPS_TEST web application, for example:

http://hips-t-app-01:12345/...

2.3.5 Post-Verification Steps

After verifying that the upgraded HIPS-Core functions as expected, perform the following additional steps:

Delete HIPS-Core queues from MSMQ.

1. Open the Computer Management console and expand Services and Applications / Message Queuing / Private Queues.
2. Delete the HIPS *ackqueue* and *hipsqueue* queues.

2.4 Upgrading HIPS UI

2.4.1 HTTPS & HTTP/2

It is strongly recommended that HIPS-UI is configured such that end-user web browsers connect to HIPS-UI over HTTPS. Depending on the server topology, the IIS instance that hosts the HIPS-UI web site components may handle HTTPS itself, or a load balancer or reverse proxy in front of one or more IIS instances may handle HTTPS and proxy web requests back to the IIS instances over either HTTPS or an unencrypted HTTP channel.

Further to this, to maximise performance for implementing sites that are using modern web browsers, HIPS-UI should be accessed using HTTPS via HTTP/2. This is provided natively by IIS 10+ (Windows Server 2016+), and is also provided by modern load balancers. HTTP/2 allows web browsers to multiplex requests on the same connection, rather than being constrained to 1 request per connection. When using HTTP/1.x, web browsers are limited to sending 2-8 requests simultaneously (web browser dependent). When using HTTP/2 the web browser can send multiple requests multiplexed on the same connection. This can significantly improve the total response time for HIPS-UI pages. It should also be noted that the use of HTTP/2 may require some additional configuration of web browser settings, which could for instance be conducted via Windows Group Policy.

2.4.2 Remove the Previous HIPS-UI Version

HIPS version 7.0.0 onward largely automates installation and removal of HIPS components via Windows PowerShell. For versions of HIPS earlier than version 7.0.0 that may be installed, it is still necessary however to manually remove HIPS components from Application Servers prior to upgrade.

IMPORTANT: Do not remove the previously created backup of the HIPS runtime folders created on each Application Server. These will be referenced in a subsequent step.

The steps required to remove a previously installed HIPS version from an Application Server depend on the version of HIPS previously installed.

To remove a version earlier than HIPS 7.0

1. Delete HIPS-UI components from IIS:
   a) Delete the HIPS-UI website.
   b) Delete the application pool used by the HIPS-UI website.
2. Delete HIPS-UI components from the filesystem:
   a) Browse to the filesystem location referenced by the previously deleted HIPS-UI website.
   b) Delete the folder and its contents.

To remove HIPS version 7.0 onwards

1. Open a Windows PowerShell console as Administrator.
2. Execute the following command to change directory to the location where the HIPS-UI installation artefacts for the previously installed HIPS version are located:

   ```
   cd <INSTALL_SOURCE_PREVIOUS_VERSION>\HIPS-Web\setup
   ```

3. Execute the following command to remove all HIPS-Web runtime components using the previously configured installation artefacts:

   ```
   .\Deploy.ps1 -Interactive -ConfigurationDataFile
   'Configuration.psd1' -Remove -Components 'Web'
   ```

   This command will invoke the `Deploy.ps1` PowerShell script using the previously configured `Configuration.psd1` file as input to remove each HIPS-UI component if already present.

4. Ensure the command completes successfully.

2.4.3 Upgrade the HIPS-UI Database

IMPORTANT:

In a High Availability topology that employs a SQL Server cluster the HIPS-UI database will be synchronised between cluster nodes.

If filesystem paths differ between cluster nodes the database files created by HIPS-UI database upgrade process may fail to be synchronised. It is recommended to defer synchronisation of the HIPS-UI database to any secondary nodes until after all steps in this section have been completed. If this is not possible, it may be necessary to temporarily remove the HIPS-UI database from synchronisation, complete the upgrade, and then re-initialise synchronisation between cluster nodes.

Locate the folder `<INSTALL_SOURCE>\HIPS-Web\database` and application named `HIPS.Web.DataStore.DBUpgrade.exe`. This application will install or upgrade the HIPS UI database objects and data into an existing database.

IMPORTANT

Ideally the login used to connect to the specified SQL Server instance must be a member of the *sysadm* fixed server role. If using integrated security, this will be the domain account of the user executing the preceding command. Alternatively, modify the connection string in the preceding command to specify the user name and password for a SQL login with the appropriate membership.

In addition, ensure the default schema of the login’s user in the HIPS UI database is set to [dbo]. It should not be set to [hipsui].

If the [SchemaVersions] table exists in the target database in a schema other than the [dbo] schema, it must be moved to the [dbo] schema prior to execution.

Open a command-prompt as Administrator and change directory to the directory which contains the `HIPS.Web.DataStore.DBUpgrade.exe` application.

Execute the following command[2]:

```
.\HIPS.Web.DataStore.DBUpgrade.exe upgrade-db "Data
Source=#{HIPS.UI.Database.Server};Initial
Catalog=#{HIPS.UI.Database.Name};Integrated Security=SSPI;Connect
Timeout=15;"
```

Modifying the highlighted values as below:

#{HIPS.UI.Database.Server} is the name of the SQL Server instance hosting the database.

#{HIPS.UI.Database.Name} is the name of the previously-created database to be upgraded.

The command uses a default command timeout of 3600 seconds (1 hour) that can be adjusted via the --timeout option if desired.

If the command was successful, a green 'Success!' notice should be displayed. Any errors will be displayed in red.

2.4.4 Configure & Install the new HIPS-UI Version

Perform the following steps on each Application Server, referencing the applicable section from the *HIPS – Initial and Clean Installation Guide (UI)* document:

Perform all steps from section *Configure Installation Artefacts*. Refer to the configuration files in the HIPS runtime folders previously backed up on each Application Server when configuring the HIPS installation artefacts.

Perform all steps from section *Execute Installation Scripts*. This will install HIPS-UI runtime components on the Application Server.

New Settings

If required, add or modify record(s) in the [hipsui].[Setting] table. Refer to the table below for more details.

Code	Setting Description
DiscloseHiddenRootFacilityCode	Disclose Hidden Health Record at a Root Facility level. If set, this is the facility code/hospital code of the seed or root organisation. Otherwise should be set to an empty string.
DefaultRemoveDocumentTab	Sets the default tab that will be displayed in HIPS UI when navigating to the Remove Document function. Should be set to one of the following values: General, Pathology, DiagnosticImaging

Click Start on the 'HIPS UI' IIS site to get it running again.

[2] Optionally, for help with additional command-line options, execute the following:
`.\HIPS.Web.DataStore.DBUpgrade.exe upgrade-db --help`

2.5 ESB changes

For pathology or diagnostic imaging sites upgrading from HIPS 6.0 or 6.1

The following changes, introduced in HIPS v6.1.2, are required to interface with the Pathology Report or Diagnostic Imaging Report HL7 interfaces of HIPS.

These changes have been applied to the Mirth Connect channel definitions supplied. If you are using Mirth Connect to interface with the Pathology Report Upload or Diagnostic Imaging Report Upload interfaces, you can install the new channel definitions, and ensure that any local changes made are merged with the new channel definitions.

- ORU messages now go to the PathologyImagingService rather than the PCEHRService
- ORU messages should be partitioned onto 40 threads using the Patient ID for partitioning
- ORU messages should be queued and retried after 10 seconds when HIPS returns a SOAP fault
- Embedded base-64 PDF data should be handled as an attachment to reduce memory usage

For sites using RefreshPatientParticipationStatus upgrading from HIPS 4.2

The RefreshPatientParticipationStatus operation was added in the PcehrServiceV2 service in HIPS 4.2 with incorrect SOAP message structure. This was corrected in later HIPS versions, and as a result there is a corresponding change needed to the integration engine. The WSDL should be re-imported on the integration engine from the upgraded HIPS system.

2.6 Finishing

You have now finished the upgrade.

Restart the relevant channels on your integration engine to resume sending HL7 messages and clinical document uploads to HIPS, and configure the load balancer to make the application available.

3 Troubleshooting

3.1 HIPS Core web services

Attempt to load the HIPS Core web service definition pages in a web browser. When on localhost, configuration errors are displayed. The port 50500 may vary.

- http://***localhost***:50500/HIPS.Service.DatabaseLoaderService.svc

If unable to load the above page, check the system Event Viewer, in particular the System and Application logs.

3.2 HIPS Core background processes

For information on troubleshooting common issues with the HIPS Core Queue Consumers refer to the *Troubleshooting* section in the *HIPS – Initial and Clean Installation Guide (Core)* document.

3.3 HIPS UI web application

Attempt to load the HIPS UI web site in a web browser. When on localhost, the ELMAH error log may be displayed at the address below. If the web site is hosted on a port other than 443, include the port.

- https://***localhost***/elmah

If unable to load the HIPS Web UI, check the dbo.ELMAH_Error table in the HIPS Web UI database and the system Event Viewer.

4 Rollback

If installation fails and it is necessary to roll back, restore the HIPS databases and binaries from backups.

Appendix A Additional Upgrade Notes

A.1 MSMQ replacement

Earlier versions of HIPS utilised Microsoft Message Queuing (MSMQ) for queuing of operations to be performed in the background. HIPS 7.0.0 onward replaces MSMQ with its own queuing mechanism implemented entirely within the HIPS Core database. Due to the significant changes required to replace MSMQ, the following guidance is provided to implementers to assist with understanding and catering for impacts as a result of this change.

A.1.1 Changes to HIPS Core Database

IMPORTANT: The *PcehrMessageQueue* table used by previous versions of HIPS has been removed from HIPS 7.0.0 onward. Its data **is not** migrated as part of the database upgrade. If it is desirable to preserve this data for reference or audit purposes, you will need to ensure it is appropriately backed up or archived elsewhere prior to upgrade.

The following changes have been made to the HIPS Core database in HIPS 7.0.0 onward as part of replacing MSMQ:

Type	**Action**	**Name**	**Comments**
Filegroup	Added	*BLOB_DATA*	
File	Added	*<HIPS-Core Database Name>*_blobs.NDF (where *<HIPS-Core Database Name>* is the name of the existing HIPS Core database)	Assigned to the new “BLOB_DATA” filegroup and located alongside the primary data file (MDF). The file may be relocated after creation if desired. The new filegroup and file are used explicitly by the new *MessageQueueContent* table (described below) to store binary message data.
Table	Added	*MessageQueue*	Replacement for the existing *PcehrMessageQueue* table & MSMQ.
Table	Added	*MessageQueueContent*	Stores binary content associated with message records in the *MessageQueue* table.
Table	Added	*MessageQueueState*	Reference entity for the state of message records in the *MessageQueue* table.
Table	Added	*MessageQueueLeaseState*	Reference entity for the state of leases on records in the *MessageQueue* table.
Table	Added	*SystemEventLog*	Stores application events. Used to provide a substitute for the *Details* column previously present in the *PcehrMessageQueue* table, as well as a foundation for richer application event logging in the future.

Type	Action	Name	Comments
Table	Added	*LogLevel*	Reference entity for the level of event log instance records in the *SystemEventLog* table.
Table	Modified	*PcehrAudit*	Added new column: *MessageQueueID* Modified to support a substitute for the *Request* and *Response* columns previously present in the *PcehrMessageQueue* table.
Table	Removed	*QueueOperation*	
Table	Removed	*PcehrMessageQueue*	
View	Added	*MessageQueueAvailable*	Supports the message queuing infrastructure.
View	Added	*MessageQueueUnavailable*	Supports the message queuing infrastructure.
View	Added	*MessageQueuePartitionedUnacquired*	Supports the message queuing infrastructure.
View	Added	*MessageQueueMetadata*	Provides a view of XML metadata associated with message records in the *MessageQueue* table.
View	Added	*SystemEventLogMetadata*	Provides a view of XML metadata associated with event log instance records in the *SystemEventLog* table.
View	Added	*SystemEventLogTags*	Provides a view of XML tags associated with event log instance records in the *SystemEventLog* table.
View	Added	*QueuedPcehrOperation*	Provides a view of data related to queued operations across a number of tables including *MessageQueue*, *MessageQueueState*, *MessageQueueLeaseState*, intended to provide a view of data similar to what was previously available in the *PcehrMessageQueue* table.
View	Added	*QueuedPcehrOperationExt*	Extends the QueuedPcehrOperation view to include data from the *MessageQueueContent*, *PcehrAudit*, *SystemEventLog* and *LogLevel* tables, intended to provide a view of data similar to what was previously available in the *PcehrMessageQueue* table.
View	Modified	*EpisodeView*	Modified to replace references to the *PcehrMessageQueue* table with the *QueuedPcehrOperation* view instead.

Type	**Action**	**Name**	**Comments**
View	Modified	*MonitoringPcehrUploadErrors*	Modified to replace references to the *PcehrMessageQueue* table with the *QueuedPcehrOperation* view instead.
Procedure	Added	*MessageQueueAcquireAvailable*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueGetCountAvailable*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueDelete*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueUpdateLease*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueUpdateState*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueInsert*	Supports the message queuing infrastructure.
Procedure	Added	*MessageQueueGet*	Supports the message queuing infrastructure.
Procedure	Added	*SystemEventLogInsert*	Supports the application event log.
Procedure	Added	*QueuedPcehrOperationSearch*	Supports querying queued operations as a substitute for several stored procedures that referenced the *PcehrMessageQueue* table.
Procedure	Modified	*PcehrAuditInsert*	Modified to support changes made to the *PcehrAudit* table.
Procedure	Modified	*MonitoringItemCounts*	Modified to replace references to the *PcehrMessageQueue* table with the *QueuedPcehrOperation* view instead.
Procedure	Modified	*MonitoringFailedUploadDocument*	Modified to replace references to the *PcehrMessageQueue* table with the *QueuedPcehrOperation* view instead.
Procedure	Modified	*ClinicalDocumentUploadList*	Modified to replace references to the *PcehrMessageQueue* table with the *QueuedPcehrOperation* view instead.
Procedure	Removed	*PcehrMessageQueueUpdate*	
Procedure	Removed	*PcehrMessageQueueOperationList*	
Procedure	Removed	*PcehrMessageQueueInsert*	
Procedure	Removed	*PcehrMessageQueueGet*	
Procedure	Removed	*PcehrMessageQueueDelete*	
Procedure	Removed	*PcehrMessageQueueCountPending*	

Of particular note, the *PcehrMessageQueue* table used by previous versions of HIPS has been replaced:

- The *MessageQueue* table replaces the *PcehrMessageQueue* table, providing a more generic message queue structure.
- Where the *PcehrMessageQueue* table provided dedicated columns for operation-specific metadata such as *EpisodeId*, *SourceSystemSetId* and *SourceSystemDocumentId*, these are now stored as XML in the *MessageMetadata* column of the *MessageQueue* table.
- If required, the *MessageQueueMetadata* view provides a rowset view of metadata extracted from the *MessageMetadata* column, for example:

MessageQueueID	Name	Value
1	EpisodeId	1
1	SourceSystemSetId	1.2.36.1.2001.1005.49.2.8003628233354453^1
1	SourceSystemDocumentId	1.2.36.1.2001.1005.49.1.8003628233354453.1805161531^1805161531

- Further, the *QueuedPcehrOperation* & *QueuedPcehrOperationExt* views provide rowset views of records from the *MessageQueue* table, metadata extracted from the *MessageMetadata* column, and several other related tables that are specific to queued operations against the My Health Record. These views may be the most appropriate replacement for implementers seeking a view of data that is equivalent to the *PcehrMessageQueue* table.

For more information on the queuing data model, refer to the *HIPS – Module Guide (Core)*.

A.1.2 Changes to HIPS Core web services

Due to database changes in HIPS 7.0.0 onward, a number of web service operations previously provided by the *PcehrService* web service in earlier versions of HIPS have been removed and replaced by substitute service operations in HIPS 7.0.0 onward:

Removed	Substitute	Migration Notes
GetIndividualOperationStatus	*GetQueuedOperation*	Supports retrieving a specific queued operation using its unique identifier. The new service operation requires the provision of a *GetQueuedOperationRequest* structure containing the unique identifier of the message representing the queued operation to be retrieved, and returns a *GetQueuedOperationResponse* structure containing the result. The behaviour of the service operation is logically equivalent to the removed service operation. The response structure is approximately equivalent to the response returned by the removed service operation, except that the *QueuedOperation* element in the response is based on the *QueuedPcehrOperation* structure described below.

Removed	Substitute	Migration Notes
GetOperationStatus	*GetQueuedOperations*	Supports retrieving a list of queued operations for a specified patient and episode. The new service operation requires the provision of a *GetQueuedOperationsRequest* structure containing the patient identifier and admission date for the episode to retrieve queued operations for, and returns a *GetQueuedOperationsResponse* structure containing the result. The behaviour of the service operation is logically equivalent to the removed service operation. The response structure is approximately equivalent to the response returned by the removed service operation, except that the *QueuedOperations* element in the response containing the set of queued operations is based on the *QueuedPcehrOperation* structure described below.
GetQueuedOperationList	*SearchQueuedOperations*	Supports retrieving a list of queued operations matching the specified criteria. The new service operation requires the provision of a *SearchQueuedOperationsRequest* structure containing the search criteria, and returns a *SearchQueuedOperationsResponse* structure containing the result. The behaviour of the service operation is logically equivalent to the removed service operation. The response structure is approximately equivalent to the response returned from the removed service operation, except that the set of queued operations in the response is based on the *QueuedPcehrOperation* structure described below.

The principal change between the removed and new service operations is the replacement in each service operation’s response of the previous *PcehrMessageQueue* and *MessageQueueItem* structures with a new *QueuedPcehrOperation* structure. This structure provides similar information to the previous structures, with the following differences noted:

PcehrMessageQueue	QueuedPcehrOperation	Migration Notes
PcehrMessageQueueId	*MessageQueueID*	
QueueOperationId	*MessageType*	The *QueueOperation* table and consequently the *QueueOperationId* foreign key has been replaced by a string value representing the type of the content of the message in the *MessageQueueContent* table. *QueueOperationId* 1 (UploadOrSupersede) now becomes *MessageType* “HIPS.PcehrSchemas.QueuedUploadOperation” *QueueOperationId* 2 (Remove) now becomes *MessageType* “HIPS.PcehrSchemas.QueuedRemoveOperation”

PcehrMessageQueue	QueuedPcehrOperation	Migration Notes
QueueStatusId	*MessageQueueStateID*	The *QueueStatus* table and consequently the *QueueStatusId* foreign key has been replaced by the *MessageQueueState* table and *MessageQueueStateID* foreign key. The values in these tables are equivalent.
SerialisedObject	*MessageContent*	
Details	*SystemEventLogMessage*	The *Details* column previously stored in the *PcehrMessageQueue* table in the case of a transient or permanent failure to process a queued operation has been replaced by the *Message* column in the new *SystemEventLog* table. Only the most recent *SystemEventLog Message* is returned.

MessageQueueItem	QueuedPcehrOperation	Migration Notes
PcehrMessageQueueId	*MessageQueueID*	
QueueItemDateCreated	*DateCreated*	
QueueOperationId	*MessageType*	The *QueueOperation* table and consequently the *QueueOperationId* foreign key has been replaced by a string value representing the type of the content of the message in the *MessageQueueContent* table. *QueueOperationId* 1 (UploadOrSupersede) now becomes *MessageType* “HIPS.PcehrSchemas.QueuedUploadOperation” *QueueOperationId* 2 (Remove) now becomes *MessageType* “HIPS.PcehrSchemas.QueuedRemoveOperation”
QueueOperationName	*N/A*	
QueueStatusId	*MessageQueueStateID*	The *QueueStatus* table and consequently the *QueueStatusId* foreign key has been replaced by the *MessageQueueState* table and *MessageQueueStateID* foreign key. The values in these tables are equivalent.
QueueStatusName	*MessageQueueStateDescription*	
Details	*SystemEventLogMessage*	The *Details* column previously stored in the *PcehrMessageQueue* table in the case of a transient or permanent failure to process a queued operation has been replaced by the *Message* column in the new *SystemEventLog* table. Only the most recent *SystemEventLog Message* is returned.
SerialisedObject	*MessageContent*	